ABTO RTP SDK

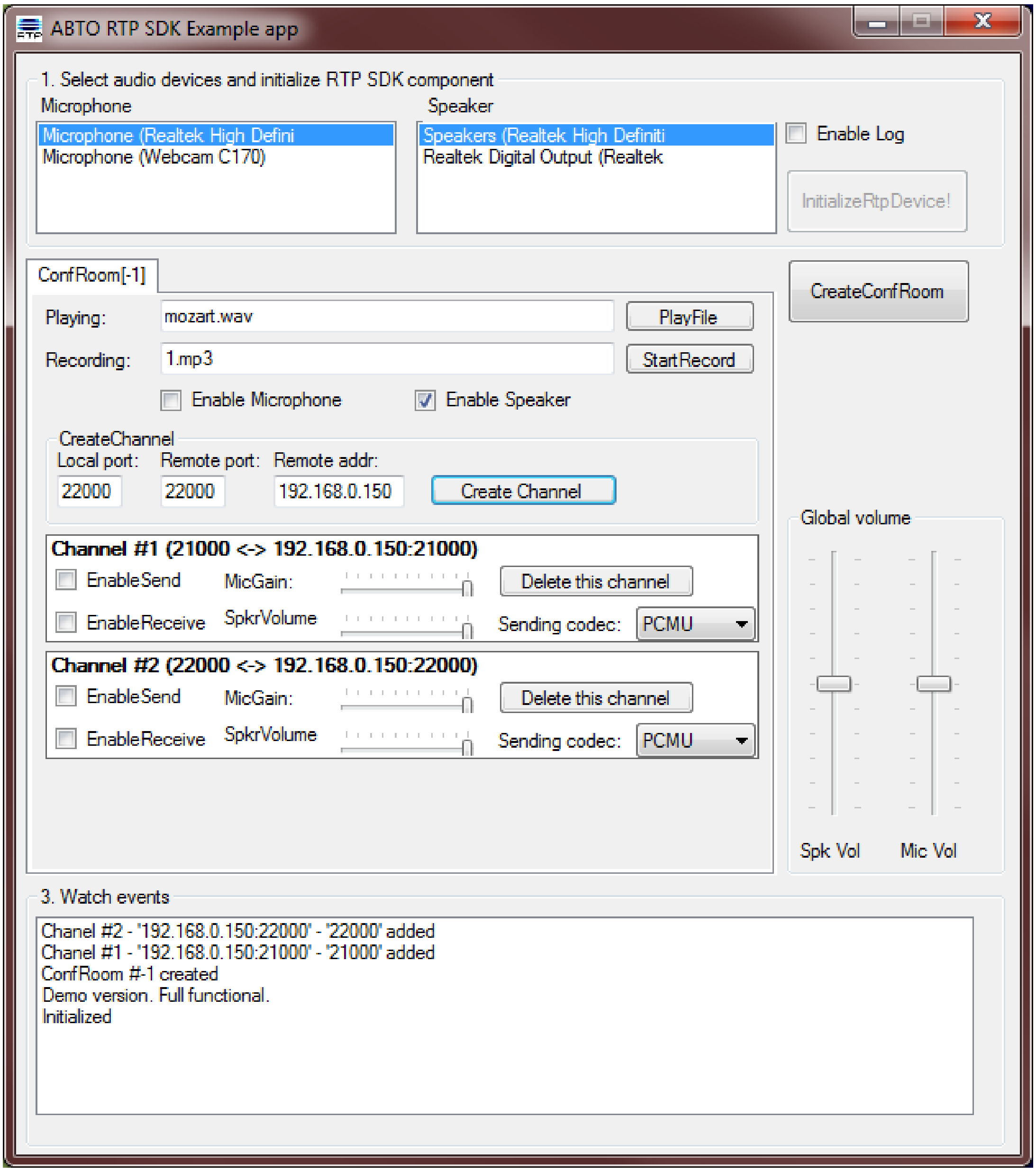
Overview

RTP SDK provides a powerful and highly customizable solution to quickly build application with ability send/receive sound by RTP protocol.

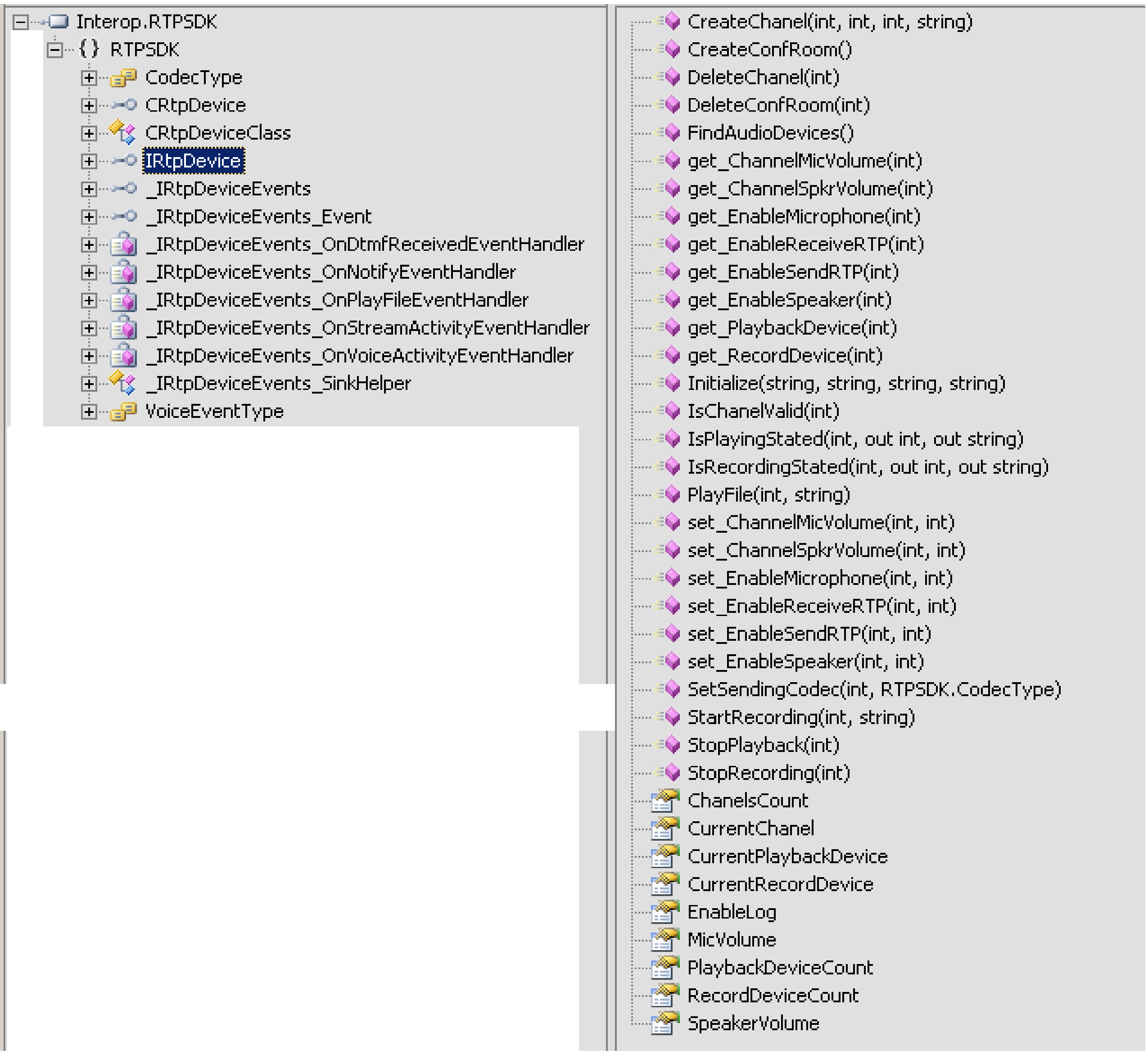
Main features:

- Ability to create multiple independent conference rooms
- Ability to create multiple channels in each conference room
- Ability to select different sending codec for each channel (G711, G726, G729, GSM, SPEEX)
- Recording in mp3 format
- Playing wav, mp3 files
- Voice detection feature

GUI of the RTP SDK example application:



RTP SDK component has following methods/properties/events:



Methods	**Description**
`CreateConfRoom`	Creates conference room. Returns id of the created room as integer. Each room has own mixer. Example: We have created conf.rom with 3 channels (A,B,C) and enabled microphone and speaker (Mic,Spkr). When channel A has enabled sending SDK will send to this channel voice from channel B and C (when receiving enabled) and voice from Mic (when enabled). It can be shown as: A = B+C+Mic B = A+C+Mic C = A+B+Mic Spkr = A+B+C
`CreateChanel`	Creates channel in conference room. Input arguments: `LONG ConfRoomId, LONG LocalPort, LONG RemotePort, String RemoteAddr` Returns id of created channel.

`DeleteChanel`	Deletes previously created channel by its id. Input argument: `LONG ChannelId`
`DeleteConfRoom`	Deletes previously created conf.room by its id. Input argument: `(LONG confRoomId)`
`FindAudioDevices`	Finds existing audio devices in system. Has to be invoked before `Initialize`, and provide ability for user selected audio devices. Note: When audio devices are not connected or not selected SDK can work without them, all local audio features will be disabled.

`Initialize`	Initializes component
	
	Input arguments:
	`String PlaybackDevice`
	`String RecordDevice –` selected playback (record) device.
	
	`String LicenseUserId`
	`String LicenseKey –` license data.

`PlayFile`
 	
`StopPlayback`
	Start/stop playing file for selected conf.room.
	
	Input arguments:
	`confRoomId` –conf.room id, received from method `CreateConfRoom`.
	
	Note:
	A. Method plays sound for all channels in the conf.room.
	B. When input file is in mp3 format SDK can automatically convert it to wav, play, remove after playing.

`IsPlayingStated`	Returns true when playing has alredy started on selected conf.room, and also returns path to file which is playing. Input arguments: `LONG ConfRoomId, out string FilePath`

`StartRecording`
 	
`StopRecording`
	Starts/stops recording file for selected conf.room.
	SDK records mixed sound from all channels and local microphone, compresses and stores in mp3 format.
	
	Input arguments:
	`confRoomId` –conf.room id, received from method `CreateConfRoom`.

`IsRecordingStated`	Returns true when recording has alredy started on selected conf.room, and also returns path to file which is recording. Input arguments: `LONG ConfRoomId, out string FilePath`
SetSendingCodec	Set codec that will be used for sending audio data. Input arguments: LONG ChanelId, CodecType codec Possible values for ‘CodecType’ are: eG711_mulaw, eG711_alaw, eGSM, eILBC, eG729A, eSPEEX, eG726

	Note: To apply new codec is required to stop and start sending again.
`set_ChannelMicVolume` `get_ChannelMicVolume`	This property allows to set input volume of the conf.room microphone which will be sent through this channel. When property set to 0 – channel will not send sound from microphone.
`set_ChannelSpkrVolume` `get_ChannelSpkrVolume`	This property allows set volume of the received voice before mix it and end to speaker. When property set to 0 – speaker will not play sound from this channel.
`set_EnableSpeaker` `get_EnableSpeaker`	Enables/Disables speaker for selected conf.room.
`set_EnableMicrophone` `get_EnableMicrophone`	Enables/Disables microphone for selected conf.room.
`set_EnableReceiveRTP` `get_EnableReceiveRTP`	Enables/Disables receiving RTP on selected channel.
`set_EnableSendRTP` `get_EnableSendRTP`	Enables/Disables sending RTP for selected channel.
SpeakerVolume	Changes global speaker volume. Input arguments: LONG Level – level in range [0..100]
MicVolume	Changes global microphone gain. Input arguments: LONG Level – level in range [0..100]

`CurrentPlaybackDevice`
`PlaybackDeviceCount`
`PlaybackDevice`
`CurrentRecordDevice`
`RecordDeviceCount`
`RecordDevice`
	Methods that returns found playback/record devices

`EnableLog`	Enables/Disables internal log features. When enabled - SDK writes file ‘RtpSdkLog.txt’ in same where component is registered. Note: this property has to be invoked before ‘Initialize’. Input arguments: `LONG Enabled` – 1-to enable log, 0 – disable.
Events	
`OnNotify`	SDK generates this event when it has some additional information about device state. Event arguments: String Msg Example: When started file playing SDK generates this event with messaga: `OnNotify("Playing started: c:\mozart.wav”)`

`OnStreamActivity`	SDK generates this event when when detected activity on some channel.
	Event arguments:
	`String Msg`
	`String Addr`
	`LONG port`
	
	Example:
	When stream started:
	`OnStreamActivity("STREAM_START", "192.168.0.56", 21000)`

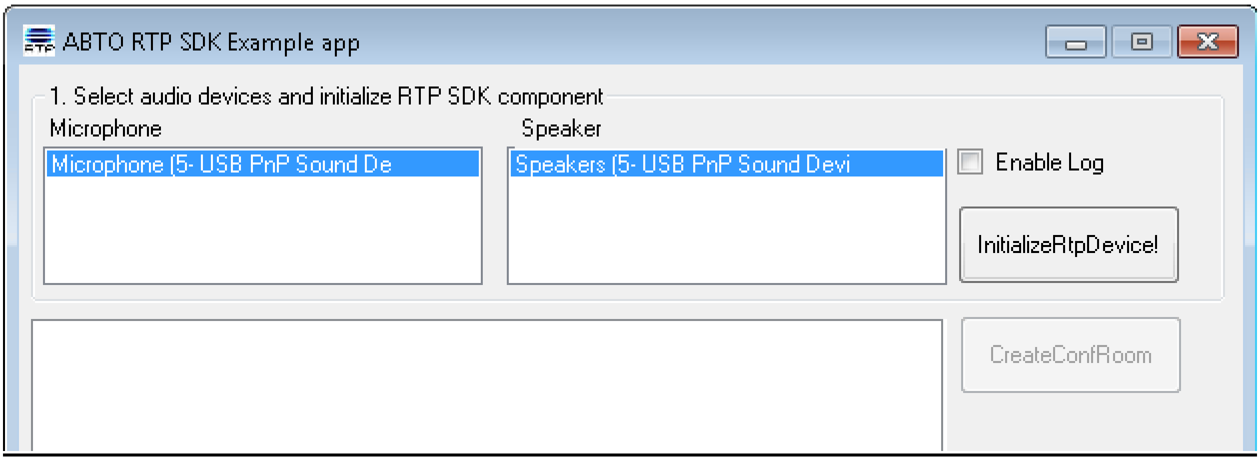
`OnPlayFile`	SDK generates this event when finished playing sound file.
	
	Event arguments:
	`VoiceEventType eventType – always equal 'eStopped'`
	`LONG ChanelId - 1-based channel id where playing finished.`

`OnVoiceActivity`	SDK generates this event when recognized some voice activity in received data. This event can be used for start/stop recording voice without silence.
	
	Event arguments:
	`VoiceEventType eventType – equal 'eStarted' when voice started and 'eStopped' when stopped.`
	`LONG ChanelId - 1-based channel id.`

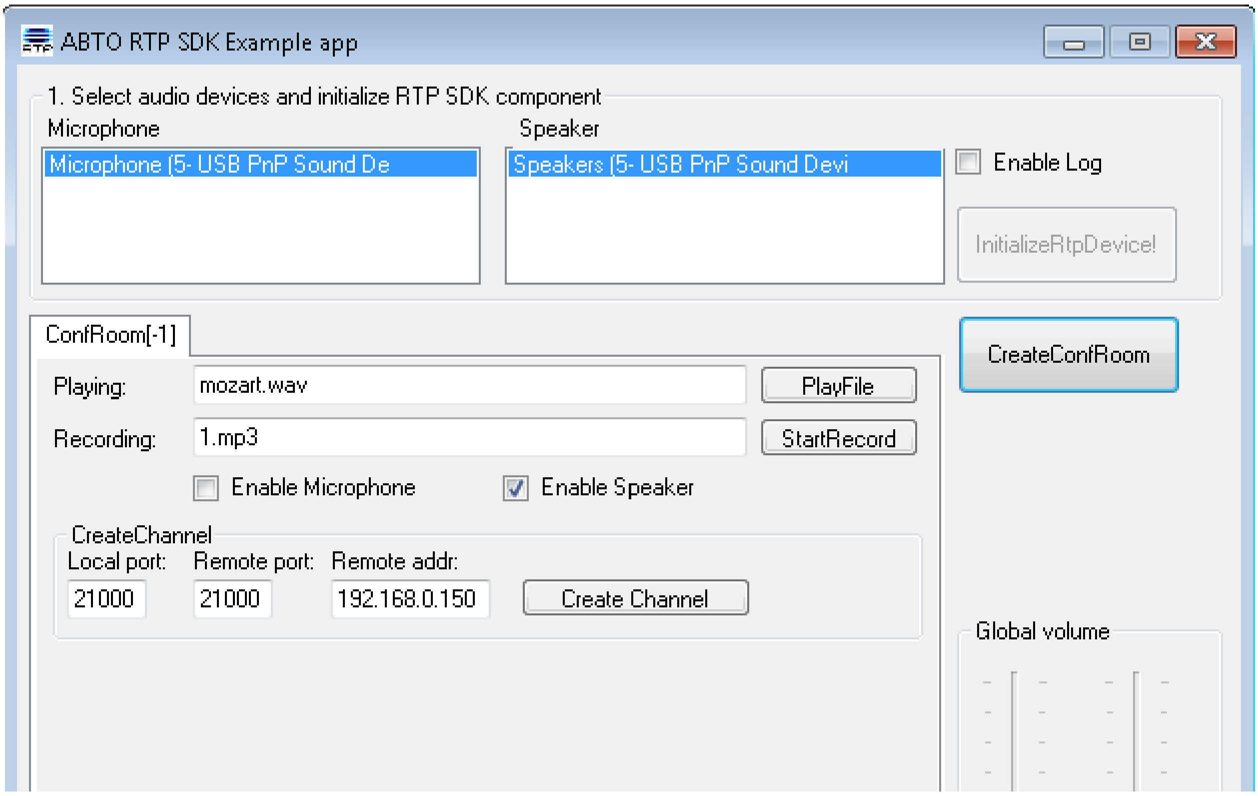
How to use example application

See source code of the example application <install_dir>\SampleWindowCS\SampleWindowCS.sln

1. Select audio devices and click "Initialize RTP device"



2. Create conf.room by clicking "CreateConfRoom" button.
   SDK/application allows create multiple conference rooms, for each room user can start playing, start recording, enable speaker and microphone. Room mixes sound from all channels as explained above in method "`CreateConfRoom`".



3. Create channels.
   Input local port, remote port and remote address and click “Create Channel”.
   Each channel independently allows enable sending/receiving.